

# A Brief Review of Hyperparameter Optimization Methods for Machine Learning

Richard Wen

*(Assignment for Research Methods in Computer Science Course at Ryerson University)*

**Abstract**—In this review, five research papers from 2015 to 2017 were selected from the ACM and IEEE Xplore digital libraries by using journal citation reports, search queries, and manual selection criteria. These five papers were summarized for a range of commonly used hyperparameter optimization methods and their performance in experimental settings. It was found that the most effective hyperparameter optimization methods used the Sequential Model Based Optimization (SMBO) framework with hyperparameter initialization techniques and incorporation of conditional spaces. However, hyperparameter optimization techniques are heavily reliant on the machine learning algorithms and datasets used, which leads to issues in dimensionality and inconsistent results. Proposed future work included implementing improved sampling techniques, and combining machine learning models with hyperparameter optimization methods for automated machine learning.

**Index Terms**—literature, review, survey, automatic, automated, hyperparameter, parameter, optimization, optimisation, machine, learning

## I. INTRODUCTION

**M**ACHINE learning is a field of computer science that studies the computer's ability to automatically learn without being specifically programmed [1]. Machine learning models have been used in important real-world applications such as identifying online criminal activity, forecasting stock markets, and diagnosing sicknesses [2], [3], [4]. Recent machine learning research has led to advances in data security, financial trading, and medical diagnoses [5], [6]. Machine learning algorithms provide a set of instructions that enable computers to learn patterns in data without being specifically programmed [7]. These patterns are used to produce models (a representation of the data learned using an algorithm) of the data where accuracies greater than 99% are possible [8]. Accuracy percentages measure the number of correctly predicted cases by the machine learning model in a given set of testing data, where values range from 0% (no cases predicted correctly) to 100% (all cases predicted correctly). Machine learning models often come with tunable values called *hyperparameters* that affect the learning performance of the underlying algorithm. Hyperparameters can be tuned to produce models that achieve high accuracy in tasks such as speech recognition and visual object detection [9], [10], [11]. Instead of inventing algorithms or sophisticated methods, the hyperparameters in machine learning algorithms are often tuned to significantly improve model performance.

One of the disadvantages of hyperparameter tuning is that it requires user expertise in the algorithms and data used to produce the machine learning model. The simplest approach is to manually explore a set of hyperparameters. This approach, however, can subjectively differ from user to user such that the resulting machine learning model becomes difficult to reproduce and scientifically justify. Machine learning models also have the disadvantage of being time-consuming to produce, and only a small range of hyperparameter choices should be explored for the sake of feasibility. In the same amount of time, optimization methods can automatically discover better hyperparameters than manual exploration. Hyperparameter optimization methods can be used to automatically select optimal hyperparameters from predefined search and time constraints. These methods are easier to theoretically justify and reproduce to verify and improve research studies and machine learning models. Hyperparameter optimization methods automate the hyperparameter selection process to reduce selection bias, and the need for expertise in particular algorithms. This allows a variety of users with different backgrounds and knowledge to apply machine learning to real-world problems. This review investigated five papers, selected using journal quality measures and digital libraries, to provide a brief summary and discussion related to hyperparameter optimization methods for machine learning algorithms.

The goal of this review was to provide general knowledge and possible research directions of recent hyperparameter optimization methods for machine learning algorithms. A brief overview and evaluation of five selected papers from the past two years were provided. The review was done by selecting relevant and recent papers using digital library search queries and manual selection criteria. The selected papers were then summarized in terms of hyperparameter optimization methods, and experimental results. A discussion of the limitations, proposed improvements, and future directions evaluated and addressed the implications of the selected papers.

The remaining sections were organized as follows:

- **Section II** detailed the methods used for digital library selection, paper selection, and reviewing selected papers
- **Section III** presented the results of the paper selection process detailed in Section II and summarizes the selected papers
- **Section IV** discussed the limitations, proposed improvements, and future directions relative to the selected papers from Section III
- **Section V** provided concluding remarks and implications

R. Wen was with the Department of Civil Engineering, Ryerson University, Toronto, ON, M5B 2K3 CA e-mail: rwen@ryerson.ca.

Assignment received October 17, 2017 and revised on November 2, 2017.

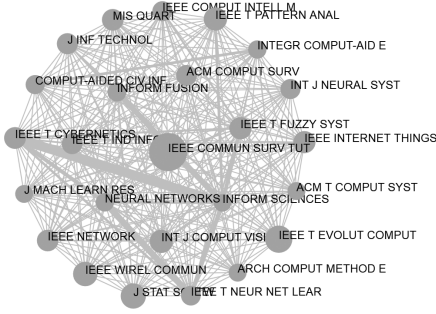


Fig. 1: **Top 25 Computer Science Journals by Journal Impact Factor from InCites Journal Citation Report in 2016.** Gray circles represent the Journal Impact Factor, where higher Journal Impact Factor values are represented by larger sizes. Connected lines represent the citation relationships between each journal, where thicker lines mean stronger relationships.

## II. METHODS

This section explains the paper selection requirements and process. Reputable digital libraries with search engines were identified by exploring journal citation reports in Section II-A. Automatic search queries from the identified digital libraries were then used to filter for potential papers in Section II-B. Section II-C details the manual selection criteria used to filter these potential papers for the selected papers in this review. The review procedure in Section II-D consisted of identifying and summarizing methods and models from the selected papers for a discussion of method and model limitations, improvements, and future directions.

### A. Digital Library Selections

The papers selected for this review were found using the search engines available in the Association for Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE) Xplore digital libraries [12], [13]. A search for the top journals in computer science by journal impact factor [14] was done using the InCites journal citation reports web tool [15] (See Appendix A). A majority of ACM and IEEE journals were found to be in the first quartile of journal impact factor values for the computer science category. A visualization of the top 25 journals in computer science by journal impact factor in 2016 is shown in Figure 1.

### B. Automatic Search Queries

Potential papers were found using search queries passed to the search engines of the ACM and IEEE Xplore digital libraries identified in Section II-A [12], [13]. Search queries were modified from the defaults and sorted by relevance (See Appendix B and C). Each search query was defined to filter for potential papers with the following requirements:

- (a) **Publication:** Published in ACM or IEEE
- (b) **Year:** Published from 2014 to October 5, 2017
- (c) **Keywords:** Contains the keywords "hyperparameter" and "optimization" in the paper title

### C. Manual Selection Criteria

The potential papers resulting from the methods in Section II-B were further filtered by inspecting the abstracts and paper length. The abstracts were inspected for relevancy to the topic: "hyperparameter optimization for machine learning". This included mentions of methods that deal with optimizing parameters for machine learning models to improve performance. After inspections of the abstract, each paper was further evaluated for practicality by searching for mentions of applications, benchmarks, and experiments in the results sections. It was assumed that papers with lengths of more than 8 pages provided adequate details and explanations of methods and results. The manual selection criteria sought to find papers with the following characteristics:

- (a) **Detailed:** Paper contained adequate details and explanations of methods and results with the assumption that papers with 8 or more pages satisfied this criteria
- (b) **Relevant:** Paper had mentions of hyperparameter optimization methods for machine learning models
- (c) **Practical:** Paper had conducted experiments, benchmarks, or applications of hyperparameter optimization methods

### D. Review Procedure

A literature review of the papers selected using the methods in Section II-C was done with the following procedure:

- 1) **Identify** methods used for hyperparameter optimization of machine learning models
- 2) **Summarize** methods in (1)
- 3) **Summarize** experiments and results for the methods in (1)
- 4) **Discuss** limitations, possible improvements, and future directions relative to the summaries from (2) and (3)

## III. RESULTS

This section details the selected papers and review procedure results described in Section II. Papers were selected from 2015 to October 5, 2017 as described in Section III-A, where a majority of the potential and selected papers were from the IEEE Xplore digital library. Summaries of the hyperparameter optimization methods, data, and experiments from the selected papers are provided in Sections III-B, and III-C respectively.

### A. Selected Papers

A total of five papers were selected for this review after applying the search queries described in Section II-B and selection criteria described in Section II-C. The search queries resulted in 14 potential papers, where 2 were from the ACM digital library, and 12 were from the IEEE Xplore digital library as seen in Figure 2. The number of potential papers and selected papers for each year from 2014 to 2017 is seen in Figure 3. The five papers selected for this review are seen in Table I, where a majority of papers were from the IEEE Xplore digital library.

TABLE I: **Selected Papers for Literature Review.** Papers were selected according to search queries described in Section II-B and selection criteria described in Section II-C. Papers are sorted in ascending order by year and then alphabetically by the first author's last name.

Authors	Title	Year	Library
Schilling N., Wistuba M., Drumond L., Schmidt-Thieme L. [16]	Joint Model Choice and Hyperparameter Optimization with Factorized Multi-layer Perceptrons	2015	IEEE
Wistuba M., Schilling N., Schmidt-Thieme L. [17]	Learning Hyperparameter Optimization Initializations	2015	IEEE
Wistuba M., Schilling N., Schmidt-Thieme L. [18]	Hyperparameter Optimization Machines	2016	IEEE
Lèvesque J-C., Durand A., Gagné C., Sabourin R. [19]	Bayesian Optimization for Conditional Hyperparameter Spaces	2017	IEEE
Quitadamo A., Johnson J., Shi X. [20]	Bayesian Hyperparameter Optimization for Machine Learning Based eQTL Analysis	2017	ACM

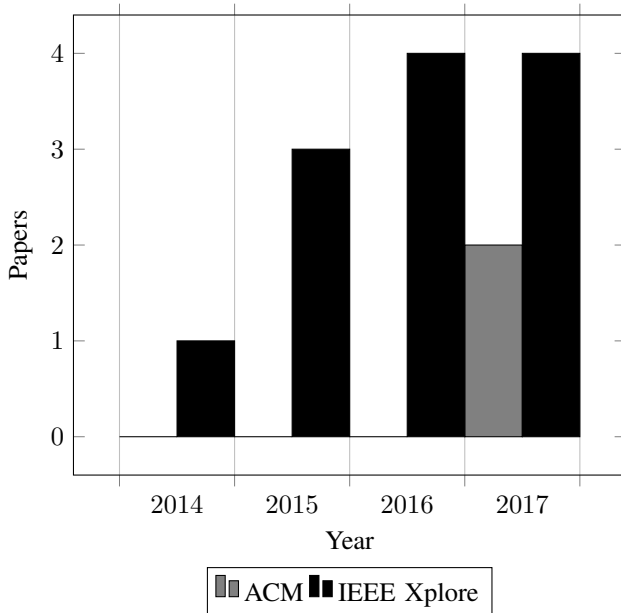


Fig. 2: **Potential ACM and IEEE Published Papers Found from 2014 to October 5, 2017.** Gray bars represent the potential papers found from the ACM digital library, and black bars represent the potential papers from IEEE Xplore digital library. Potential papers were found with search queries as described in Section II-B.

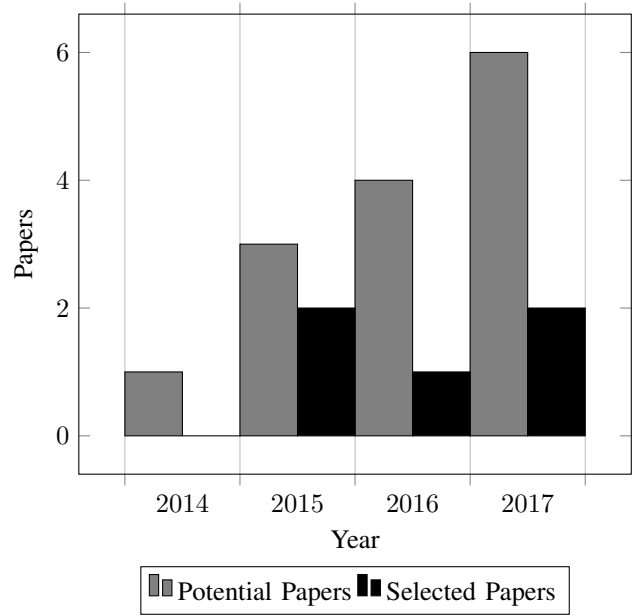


Fig. 3: **ACM and IEEE Published Papers Found from 2014 to October 5, 2017.** Gray bars represent the potential papers found using the search queries described in Section II-B. Black bars represent the selected papers filtered from the potential papers using the selection criteria described in Section II-C.

### B. Hyperparameter Optimization Methods

Hyperparameter optimization methods search a set of machine learning parameters to maximize a performance measure, where higher values represent better performance.

A hyperparameter optimization method seeks to find a set of parameters  $x_1 \dots x_n$  that maximize the performance measure function  $f$  (often time-consuming to evaluate) as seen in Equation 1:

$$\{x_1 \dots x_n\} \text{ such that } \arg \max_{x_1 \dots x_n} f(x_1 \dots x_n) \quad (1)$$

The set of parameters  $x_1 \dots x_n$  is then a search space of possible user inputs to a machine learning model. The idea is to create constraints (manually done by human intervention) on  $x_1 \dots x_n$  by limiting the range of values, number of iterations, or allowed running time [11]. A hyperparameter optimization method can then be used to select the best  $x_1 \dots x_n$  based on these constraints.

Simple hyperparameter optimization methods mentioned in the selected papers included manual search, grid search, and random search [21]. A manual search requires the user to use a trial and error process to find a selection of parameter values on a machine learning algorithm that results in generally adequate performance. The user is usually guided by expertise on the algorithm and dataset or through external knowledge such as previous experiments, literature, or other relevant experts. Grid search involves limiting the search space of parameter values to a predefined range of values to search in. The predefined range of values is based on the discretion of the user, which requires knowledge of appropriate ranges to

TABLE II: Simple Hyperparameter Optimization Methods.

Method	Description
Manual Search	Trial and error attempt to find optimal hyperparameters
Grid Search	Predefined range of values to limit optimal hyperparameter search space
Random Search	Randomized range of values to limit optimal hyperparameter search space

use for the machine learning algorithm and dataset. Random search is similar to grid search, except that the predefined range of values are randomized according to a probability distribution. The randomization process enables larger ranges to be defined as not all values in the range will be used, which reduces the time required to optimize the machine learning hyperparameters. Simple hyperparameter optimization methods (summarized in Table II) are easy to implement and their effectiveness are based on the user's expertise of the machine learning algorithms and datasets chosen.

Advanced hyperparameter optimization methods in the selected papers included assumption based optimization, evolutionary based optimization, and Sequential Model Based Optimization (SMBO). Assumption based optimization creates particular hyperparameter models for particular algorithms, datasets, and situations [16]. Evolutionary based optimization uses evolutionary algorithms, inspired by biology, to search for optimal hyperparameters in machine learning models [22]. These evolutionary algorithms produce several machine learning models varied by hyperparameter values, evolves these produced models by mutation, and selects the best performing evolved models. SMBO creates a model of the performance measure function, treated as a black-box function, to predict the next best set of hyperparameters to evaluate [23]. Gaussian models (Bayesian optimization) [24] and random forest models [25] have been used in SMBO to model the performance measure function in order to find optimal hyperparameters. Advanced hyperparameter optimization methods (summarized in Table III) provide more efficient and effective approaches than simple hyperparameter optimization methods under the same constraints such as number of iterations or allowed running time.

Several improvements to SMBO have been proposed by the authors of the selected papers. Schilling, Wistuba, Drumond, and Schmidt-Thieme [16] used factorized multilayer perceptions to form lower dimensional spaces. The integration of the factorized multilayer perceptron model into the SMBO allowed SMBO to search for optimal hyperparameters across different datasets and machine learning algorithms. Wistuba, Schilling, and Schmidt-Thieme [17] used an initialization learning algorithm to transfer more optimal initial hyperparameter values from previous experiments. This enabled more efficient searching by providing more optimal starting values for the search space in hyperparameter optimization methods. Wistuba, Schilling, and Schmidt-Thieme [18] proposed an improved framework over SMBO called Hyperparameter

TABLE III: Advanced Hyperparameter Optimization Methods.

Method	Description
Assumption Based	Specific hyperparameter models based on expert assumptions of particular algorithms and datasets
Evolutionary Based	Models that attempt to select the best performing hyperparameters from a set of procedurally generated hyperparameters
Sequential Model Based	Models of the performance measure function to sequentially guide the next hyperparameters to be evaluated

TABLE IV: Sequential Model Based Optimization Improvements.

Authors	Method	Improvement
Schilling N., Wistuba M., Drumond L., Schmidt-Thieme L. [16]	Factorized multilayer perceptron	Enabled hyperparameter optimization across different datasets and machine learning algorithms
Wistuba M., Schilling N., Schmidt-Thieme L. [17]	Hyperparameter initialization learning	Provided more optimal starting points in hyperparameter search space
Wistuba M., Schilling N., Schmidt-Thieme L. [18]	Hyperparameter Optimization Machines (HOM)	Provided a way to measure the influence of transferred hyperparameter values from previous experiments
Lèvesque J-C., Durand A., Gagné C., Sabourin R. [19]	Conditional kernels	Provided a method to optimize for hyperparameters that are dependent on other hyperparameters or have the ability to be inactive

Optimization Machines (HOM) that used a transfer function in addition to the original model in SMBO. The transfer function controlled the influence of transferred hyperparameter knowledge from previous experiments such that hyperparameter values can become irrelevant once enough data about the search space is gathered. Lèvesque, Durand, Gagné, and Sabourin [19] proposed conditional kernels for Bayesian optimization (a SMBO method) that handled conditional hyperparameter spaces with Gaussian Processes. Conditional hyperparameter spaces occurred when there were hyperparameters that were dependent on other hyperparameters or had the ability to be inactive (having no value). A summary of the improvements to SMBO are provided in Table IV.

### C. Data and Experiments

The selected papers ran experiments on the several hyperparameter optimization methods to evaluate and compare their performance:

1) *Schilling, Wistuba, Drumond, and Schmidt-Thieme (2015)*: Schilling et al. [16] performed experiments (publicly

available at <http://www.hylap.org>) for 59 created datasets (from the Waikato Environment for Knowledge Analysis (WEKA) [26]) with 1873 subsample hyperparameter configurations for 4 hyperparameter optimization methods. The results revealed that factored multilayer perceptrons models achieved greater normalized accuracy (90% versus 70%) for the same machine learning algorithms at earlier points in time by learning initializations from meta-data. Meta-data contain data that provides abstract information (such as titles, dates, and locations) and structural formats (such as file formats and column descriptions) on other data, which can improve the performance of hyperparameter optimization algorithms.

2) *Wistuba, Schilling, and Schmidt-Thieme (2015)*: Wistuba et al. [17] performed experiments that compared hyperparameter optimization methods with and without initialized hyperparameter values based on 80% training and 20% testing splits of 50 classification datasets (from A Library for Support Vector Machines (LIBSVM) [27] and MultiBoost [28]). The results revealed that the hyperparameter optimization methods with initialized hyperparameter values performed better due to initialization learning from previous experiments.

3) *Wistuba, Schilling, and Schmidt-Thieme (2016)*: Wistuba et al. [18] later performed experiments on two different meta-data sets (generated by LIBSVM [27] and WEKA [26] with 50 datasets and 59 datasets respectively) for hyperparameter optimization and combined algorithm selection. The results revealed that a HOM framework with Adaptive Hyperparameter Transfer Learning with Simple Surrogates (AHT) provided the best results in terms of space and time compared to regular initialization with SMBO.

4) *Lèvesque, Durand, Gagnè, and Sabourin (2017)*: Lèvesque et al. [19] performed experiments on hyperparameter optimization methods with and without conditional kernels on 24 datasets (extracted from the OpenML repository [29] and University of California Irvine (UCI) machine learning repository [30]) that consisted of 500 to 60000 instances. The results revealed that the hyperparameter optimization methods with conditional kernels outperformed other optimization methods that did not use conditional kernels on the datasets.

5) *Quitadamo, Johnson, and Shi (2017)*: Quitadamo, Johnson, and Shi [20] performed experiments of hyperparameter optimization on large genomic datasets (extracted from the 1000 Genomes project [31] and Genetic European Variation in Health and Disease (gEUVADIS) project [32]), where the number of columns are significantly larger than the number of rows, for the methods of grid search, random search, and Bayesian optimization. The results revealed that Bayesian optimization found better parameters than grid search or random search in the same amount of time for the genomic dataset.

The datasets used for the mentioned experiments used publicly available datasets generated from popular machine learning libraries and repositories as seen in Table V (with the exception of Quitadamo et al. [20]). The hyperparameter optimization methods that had high performance incorporated the SMBO framework with initialization of hyperparameter values from transfer learning techniques.

TABLE V: Datasets Used in Selected Papers.

Dataset	# Papers
1000 Genomes project [31]	1 [20]
A Library for Support Vector Machines (LIBSVM) [27]	2 [17] [18]
Genetic European Variation in Health and Disease (gEUVADIS) project [32]	1 [20]
MultiBoost [28]	1 [17]
OpenML [29]	1 [19]
University of California Irvine (UCI) Machine Learning Repository [30]	1 [19]
Waikato Environment for Knowledge Analysis (WEKA) [26]	2 [16] [18]

#### IV. DISCUSSION

This section discusses the limitations, proposed improvements, and future directions for the selected papers in Table I. Section IV-A discusses the limitations of hyperparameter optimization relative to dimensionality, dataset characteristics, and machine learning algorithm choice. Section IV-B discusses the proposed improvements and future research directions relative to sampling, model combination, and automated machine learning.

##### A. Limitations

The limitations of hyperparameter optimization methods for machine learning algorithms include issues with high dimensional hyperparameters, large datasets, costly performance measure functions, and variations between datasets. High dimensional hyperparameters lead to longer evaluation times for the performance measure function, which may noticeably lower the plausible search space. Extremely large datasets, similar to high dimensional hyperparameters, also increase the evaluation times for the performance measure function, which can drastically reduce the plausible search space. The running time constraint is also heavily influenced by the time expense of the performance measure functions. If the performance measure functions are not scalable, hyperparameter optimization methods cannot gather enough data in feasible times to produce an adequate model. Hyperparameter optimization methods also vary between different datasets, and there does not exist one method that will provide the best results in every scenario. Although hyperparameter optimization methods are effective for producing machine learning models with high performance, the methods are not entirely automatic (human intervention is required to select the parameter ranges, and time constraints) and are heavily reliant on the machine learning algorithms and the datasets being used.

##### B. Improvements and Future Directions

The improvements and future directions of hyperparameter optimization methods for learning algorithms include sampling

techniques, combination of machine learning models, and automated machine learning. Results from hyperparameter optimization are highly dependent on the datasets and machine learning algorithms used. The dimensionality of the dataset and hyperparameters can affect the plausibility of running optimization methods. Sampling techniques can help reduce the dimensionality and the size of datasets and hyperparameters in order to improve running times, and time constraints on hyperparameter optimization methods. The combination of machine learning models and algorithms should be considered to produce new machine learning models that have the potential to outperform non-combined models. The process of combining machine learning models through hyperparameter optimization methods may lead into the direction of automated machine learning (notable examples include Auto-WEKA [33], Tree Based Pipeline Optimization Tool (TPOT) [34], and auto-sklearn [35]), where new models can be created and adjusted without the need for user expertise.

## V. CONCLUSION

This review selected papers from the ACM and IEEE Xplore digital libraries by evaluating journal citation reports, using automatic search queries, and applying manual selection criteria. The review methods limited the papers to content relevant to hyperparameter optimization methods for machine learning for the years 2015 to 2017. The five selected papers were summarized for optimization methods, improvements, and experiments for an overview of the recent advances in hyperparameter optimization. Sequential Model Based Optimization (SMBO) was a common framework used among all papers, where comparisons and improvements from each study were made to SMBO. Improvements to SMBO included the use of hyperparameter initialization techniques [16], [17], a Hyperparameter Optimization Machines (HOM) framework [18], and conditional hyperparameter spaces [19]. The limitations of hyperparameter optimization methods included the size of datasets and the effectiveness of machine learning algorithms to that dataset. These limitations pose issues in requiring some human selection bias (such as choosing constraints for parameter ranges, iterations, or time) that can lead to feasibility problems and inconsistent results. Sampling was proposed as a possible solution to combat the limitations of high dimensionality in the datasets and hyperparameters. The application of hyperparameter optimization to the process of combining machine learning models was proposed as a way to automatically create new machine learning models. Research directions in hyperparameter optimization methods include the improvement of sampling techniques and the application of hyperparameter optimization to automated machine learning.

## APPENDIX A INCITES SEARCH

The search for the top computer science journals based on Journal Impact Factor (JIF) [14] was done using the InCites Journal Citation Reports (JCR) web tool [15]. The search used the following options available on InCites:

- **Categories:**

- COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE
- COMPUTER SCIENCE, CYBERNETICS
- COMPUTER SCIENCE, HARDWARE & ARCHITECTURE
- COMPUTER SCIENCE, INFORMATION SYSTEMS
- COMPUTER SCIENCE, INTERDISCIPLINARY APPLICATIONS
- COMPUTER SCIENCE, SOFTWARE ENGINEERING
- COMPUTER SCIENCE, THEORY & METHODS

- **JCR Year:** 2016

- **Edition:** Science Citation Index Expanded (SCIE) [36] and Social Sciences Citation Index (SSCI) [37]

- **Category Schema:** Web of Science [38]

- **JIF Quartile:** Quarter 1 (Q1)

## APPENDIX B ACM QUERY SYNTAX

The query syntax in the ACM digital library was accessed through the advanced search page by clicking "*show query syntax*". The "+" symbol includes each keyword in the title. "*gte*" and "*lte*" represent "*greater than or equal to*" and "*less than or equal to*" respectively. A filter for the publication year must be applied manually through the web interface to generate the appropriate query syntax. The advanced query syntax used for the ACM digital library to return potential papers is shown below:

```
"query": { acmdlTitle:(+hyperparameter +optimization) }
"filter": { "publicationYear":{ "gte":2014, "lte":2017 }},
{owners.owner=HOSTED}
```

## APPENDIX C IEEE XPLORE COMMAND SEARCH

The command search in the IEEE Xplore digital library was accessed through the advanced search page by clicking "*command search*". Refinements were manually applied using the web interface to filter command search results for the years 2014 to 2017 and to search in "*Metadata only*". The command search used for the IEEE Xplore digital library to return potential papers is shown below:

```
("Document Title":hyperparameter)
AND
("Document Title":optimization)
```

## ACKNOWLEDGEMENTS

I would like to thank Dr. Cherie Ding for providing the assignment outline, resources, directions, paper selection advice, and feedback for this literature review.

## REFERENCES

- [1] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.
- [2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [3] J. Heaton, N. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.
- [4] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf *et al.*, "Kernel mean embedding of distributions: A review and beyond," *Foundations and Trends® in Machine Learning*, vol. 10, no. 1-2, pp. 1–141, 2017.
- [5] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [6] M. de Bruijne, "Machine learning approaches in medical image analysis: From detection to diagnosis."
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
- [9] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.
- [10] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.
- [11] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 2546–2554.
- [12] Association for Computing Machinery. (2017) Acm digital library. Accessed: 2017-10-05. [Online]. Available: <https://dl.acm.org/>
- [13] Institute of Electrical and Electronics Engineers. (2017) Ieee xplora digital library. Accessed: 2017-10-05. [Online]. Available: <http://ieeexplore.ieee.org/Xplore/home.jsp>
- [14] E. Garfield, "The history and meaning of the journal impact factor," *Jama*, vol. 295, no. 1, pp. 90–93, 2006.
- [15] Clarivate Analytics. (2017) Incites journal citation reports. Accessed: 2017-10-05. [Online]. Available: <https://jcr.incites.thomsonreuters.com/>
- [16] N. Schilling, M. Wistuba, L. Drumond, and L. Schmidt-Thieme, "Joint model choice and hyperparameter optimization with factorized multi-layer perceptrons," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov 2015, pp. 72–79.
- [17] M. Wistuba, N. Schilling, and L. Schmidt-Thieme, "Learning hyperparameter optimization initializations," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2015, pp. 1–10.
- [18] —, "Hyperparameter optimization machines," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2016, pp. 41–50.
- [19] J. C. Lèvesque, A. Durand, C. Gagné, and R. Sabourin, "Bayesian optimization for conditional hyperparameter spaces," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 286–293.
- [20] A. Quitadamo, J. Johnson, and X. Shi, "Bayesian hyperparameter optimization for machine learning based eqtl analysis," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ser. ACM-BCB '17. New York, NY, USA: ACM, 2017, pp. 98–106. [Online]. Available: <http://doi.acm.org/10.1145/3107411.3107434>
- [21] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.
- [22] D. Whitley, S. Rana, J. Dzuber, and K. E. Mathias, "Evaluating evolutionary algorithms," *Artificial intelligence*, vol. 85, no. 1, pp. 245–276, 1996.
- [23] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [24] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.
- [25] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," *LION*, vol. 5, pp. 507–523, 2011.
- [26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [27] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [28] D. Benbouzid, R. Busa-Fekete, N. Casagrande, F.-D. Collin, and B. Kégl, "Multiboost: a multi-purpose boosting package," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 549–553, 2012.
- [29] J. Vanschoren, J. N. Van Rijn, B. Bischl, and L. Torgo, "Openml: networked science in machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49–60, 2014.
- [30] M. Lichman. (2013) Uci machine learning repository. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [31] P. H. Sudmant, T. Rausch, E. J. Gardner, R. E. Handsaker, A. Abyzov, J. Huddleston, Y. Zhang, K. Ye, G. Jun, M. H.-Y. Fritz *et al.*, "An integrated map of structural variation in 2,504 human genomes," *Nature*, vol. 526, no. 7571, pp. 75–81, 2015.
- [32] T. Lappalainen, M. Sammeth, M. R. Friedländer, P. Act Hoen, J. Monlong, M. A. Rivas, M. Gonzalez-Porta, N. Kurbatova, T. Griebel, P. G. Ferreira *et al.*, "Transcriptome and genome sequencing uncovers functional variation in humans," *Nature*, vol. 501, no. 7468, pp. 506–511, 2013.
- [33] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-weka: Combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 847–855.
- [34] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO '16. New York, NY, USA: ACM, 2016, pp. 485–492. [Online]. Available: <http://doi.acm.org/10.1145/2908812.2908918>
- [35] M. Feuer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems*, 2015, pp. 2962–2970.
- [36] E. Garfield, "Citation indexes for science. a new dimension in documentation through association of ideas," *International journal of epidemiology*, vol. 35, no. 5, pp. 1123–1127, 2006.
- [37] D. B. Klein, E. Chiang *et al.*, "The social science citation index: A black box with an ideological bias?" *Econ Journal Watch*, vol. 1, no. 1, pp. 134–165, 2004.
- [38] Clarivate Analytics. (2017) Web of science. Accessed: 2017-10-05. [Online]. Available: <https://webofknowledge.com/>